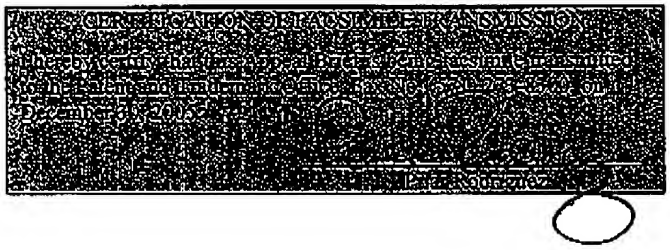
Patent

Attorney Docket No.: Intel 2207/7085
Serial No.: 09/749,405
Assignee: Intel Corporation

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT	:	STEPHAN J. JOURDAN et al.
SERIAL NO.	:	09/749,405
FILING DATE	:	December 28, 2000
GROUP ART UNIT	:	2183
FOR	:	METHOD AND APPARATUS FOR PREDICTING BRANCHES USING A META PREDICTOR
EXAMINER	:	Aimee J. Li

M/S: APPEAL BRIEFS - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450



APPEAL BRIEF

Dear Sir:

This brief is in furtherance of the Notice of Appeal, filed in this case on August 31, 2005.

1. REAL PARTY IN INTEREST

The real party in interest in this matter is Intel Corporation. (Assignment recorded April 23, 2001, Reel/Frame 011786/0522).

01/04/2006 EFLORES 00000018 110600 09749405
02 FC:1402 500.00 DA

2. RELATED APPEALS AND INTERFERENCES

There are no related appeals.

3. STATUS OF THE CLAIMS

Claims 1-27 are pending in this application. Claims 1-8, 10-15 and 17-27 were rejected under 35 U.S.C. §103(a).

4. STATUS OF AMENDMENTS

The claims listed on page 1 of the Appendix attached to this Appeal Brief reflect the present status of the claims.

5. SUMMARY OF THE CLAIMED SUBJECT MATTER

The embodiment of claim 1 generally describes a branch prediction apparatus, comprising: a base misprediction history register (e.g., *see* page 9, lines 9-10) to provide an output; a meta predictor (e.g., *see* page 4, lines 21-23 and Figure 2, 100) to receive as inputs an index value (e.g., *see* page 4, line 24 and Figure 2, 106) and a branch prediction (e.g., *see* page 4, line 24 and Figure 2, 108) to generate a misprediction value (e.g., *see* page 4, line 25 and Figure 2, 112) in accordance with said inputs and said base misprediction history register output; and a logic gate (e.g., *see* page 8, line 15 and Figure 3, 214) to receive said branch prediction and said misprediction value to generate a final prediction (e.g., *see* page 8, lines 20 and Figure 3, 216).

The embodiment of claim 10 generally describes a method for predicting branches, comprising: receiving an index value (e.g., *see* page 4, line 24 and Figure 2, 106), a branch prediction value correlating to said index value (e.g., *see* page 4, line 24 and Figure 2, 108), and a misprediction history value (e.g., *see* page 4, line 25 and Figure 2, 112) at a meta predictor (e.g., *see* page 4, lines 21-23 and Figure 2, 100); and generating a misprediction value (e.g., *see* page 4, line 25 and Figure 2, 112) at said meta predictor.

The embodiment of claim 17 generally describes a processor, comprising: a branch predictor (e.g., *see* page 4, line 24 and Figure 2, 108) to generate a branch prediction (e.g., *see* page 4, line 24 and Figure 2, 108); a base misprediction history register (e.g., *see* page 9, lines 9-10); a meta predictor (e.g., *see* page 4, lines 21-23 and Figure 2, 100) to receive an index value (e.g., *see* page 4, line 24 and Figure 2, 106), said branch prediction and base misprediction history register data to generate a misprediction value.

The embodiment of claim 20 generally describes a computer readable medium having stored a plurality of executable instructions, the plurality of instructions comprising instructions to: receive an index value (e.g., *see* page 4, line 24 and Figure 2, 106), a branch prediction value (e.g., *see* page 4, line 24 and Figure 2, 108) correlating to said index value, and a misprediction history value (e.g., *see* page 4, line 25 and Figure 2, 112) at a meta predictor (e.g., *see* page 4, lines 21-23 and Figure 2, 100); and generate a misprediction value at said meta predictor (e.g., *see* page 4, line 25 and Figure 2, 112).

The embodiment of claim 24 generally describes a method for restoring a branch prediction apparatus following a branch misprediction of a branch instruction, comprising:

restoring a base misprediction history register (e.g., *see* page 8, lines 22-24); and restoring a branch predictor history register (e.g., *see* page 8, lines 22-24).

Referring now in detail to the drawings wherein like parts are designated by like reference numerals throughout, Fig. 2 depicts a meta predictor apparatus 100 according to an embodiment of the present invention. Meta predictor apparatus 100 includes meta predictor 104. Meta predictor 104 receives index information 106 and branch prediction 108 and reads from base misprediction history register 110 in generating misprediction value 112. Final value 118 is the final value after the instruction has been processed, and the branch resolved. If branch prediction 108 is correct, then it should equal final value 118. If meta predictor apparatus 100 predicts that branch prediction 108 is incorrect, then meta predictor 104 generates misprediction value 112 to alter branch prediction 108. This altered value is used as the predicted outcome of the branch instruction, and should be compared to final value 118 at the execution stage.

The base misprediction history register 110 may be updated by comparing the final value 118 to the branch prediction 108. Base misprediction history register 110 reflects whether the last N instances of branch prediction 108 have been correct or incorrect. If branch prediction 108 does not equal final value 118, then base misprediction history register 110 inserts a 1 by shifting the register. If branch prediction 108 equals final value 118, then the original branch prediction is correct and base misprediction history register 110 inserts a 0 by shifting the register. Thus, clusters of mispredictions are stored. Further, because mispredictions tend to occur in clusters, base misprediction history register 110 should include at least one misprediction prior to carrying out any meta predictions via meta predictor 104. If base

misprediction history register 110 is nothing but 0 values, then meta predictor 104 would not reverse any branch prediction 108. Meta predictor 104 should be bypassed, and not be accessed, if base misprediction history register 110 is filled with 0 values. A transition should occur before meta predictor 104 is brought into the prediction operations for branch prediction 108. Thus, in cases where base misprediction history register 110 is all 0 values, branch prediction 108 may not be reversed or altered by meta predictor 104.

Meta predictor 104 also reads misprediction history data 114 from base misprediction history register 110. Using misprediction history data 114, branch prediction 108 and index value 106, meta predictor 104 generates misprediction value 112. Misprediction value 112 then may be used to decide whether to reverse the prediction provided by the base predictor, or branch prediction 108. Meta predictor 104 may implement any scheme suitable for binary prediction. For example, meta predictor 104 can implement a two level prediction scheme, much like most branch predictors.

Fig. 3 depicts a branch prediction apparatus 200 according to an embodiment of the present invention. Branch predictor 202 is coupled to meta predictor 104. Branch predictor 202 generates branch prediction 108 according to a prediction scheme implemented by branch predictor 202. Branch predictor 202 receives index value 106. As discussed above, index value 106 may be any information used by the prediction scheme in branch predictor 202. Prediction schemes use index value information in determining whether a branch should be taken or not taken. For example, index value 106 may be an instruction pointer address for the address of the

branch instruction. Branch prediction 108 is a taken/not taken prediction generated by branch predictor 202.

Misprediction value 112 is used to augment branch prediction 108. Logic gate 214 receives misprediction value 112 and branch prediction 108. Logic gate 214 determines whether to reverse branch prediction 108 according to misprediction value 112. If misprediction value 112 predicts that branch prediction 108 is correct, then logic gate 214 does not reverse branch prediction 108. If misprediction value 112 predicts that branch prediction 108 in incorrect, then logic gate 214 reverses branch prediction 108. Using the inputs, logic gate 214 generates a final prediction 216. Final prediction 216 predicts whether the branch instruction should be taken or not taken.

Fig. 4 depicts a flowchart of a method for predicting branches according to an embodiment of the present invention.

Fig. 5 depicts a flowchart of a method for resolving a branch misprediction within a branch predictor and a meta predictor in accordance with an embodiment of the present invention.

6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Are claims 24-26 anticipated under 35 U.S.C. 102(b) as being taught by Tran (U.S. Patent No. 5,822,575).

B. Are claims 1-8, 10-15, and 17-23 rendered obvious under 35 U.S.C. 103(a) under Chang, Hao, and Patt's "Alternative Implementations of Hybrid Branch Predictors" (herein referred to as "Patt") in view of McFarling's "WRL Technical Note TN-36: Combining Branch Predictors" (herein referred to as "McFarling")?

7. ARGUMENT

B. Claims 1-8, 10-15, and 17-23 are not obvious under 35 U.S.C. 103(a) under Patt in view McFarling.

The Examiner asserts that the limitation "...[a] branch prediction apparatus, comprising:...a meta predictor to receive as inputs an index value and a branch prediction to generate a misprediction value in accordance with said inputs and..." (e.g., as described in the embodiment of claim 1), can be found in Patt at page 252, column 2, paragraph 4; page 255, column 1, paragraph 2 – column 2, paragraph 3; and Figure 2. Applicant respectfully disagrees.

Column 2, paragraph 4 of Patt discloses:

> Hybrid branch predictors were first proposed by McFarling [6]. They consist of a set of single-scheme predictors and a prediction selection mechanism. For each branch, each of the single-scheme predictors makes a prediction. The selection mechanism then chooses one of the predictions to be the hybrid predictor's prediction. McFarling proposed implementing the selection mechanism as an array of 2-bit counters. Each static branch was associated a counter which would keep track of which predictor was currently more accurate for that branch. We will refer to this array as the branch predictor selection table (BPST). Upon confirmation of a branch prediction, the counter would be incremented or decremented depending on which single-scheme predictor was correct. If both were correct (or incorrect), the counter state would be left unchanged.

Applicants submit column 2, paragraph 4 of Patt is intended to disclose a selection mechanism used amongst a set of single scheme predictors. Specifically, each single scheme predictor makes a prediction, from which the selection mechanism would choose. It further discloses a counter *to keep track of which predictor selection was currently more accurate.* It does so by simply incrementing or decrementing the counter based on which of the single scheme predictors were correct. Therefore, this Patt counter and any value associated with does not actually lead to or factor into any prediction, but merely keeps track of an overall accuracy of a group of predictors. In other words, any value associated with the counter does not have any active, determinative purpose (like for example, a misprediction value), but rather is merely used to

passively reflect the overall effectiveness of the prediction scheme. Therefore, Applicants submit that this "counter" and any associated counter cannot be the equivalent of a "meta predictor to receive as inputs an index value and a branch prediction *to generate a misprediction value*" (e.g., as described in the embodiment of claim 1).

Column 1 paragraph 2 to column 2, paragraph 3 of page 255 discloses:

> 4. Predictor Selection Mechanisms
>
> The performance of a hybrid branch predictor also depends on its predictor selection mechanism. In the previous section, we have determined the optimal configurations of hybrid branch predictors with an idealized static selection mechanism. In this section, we will use these same configurations for evaluating the performance of our real selection mechanisms. We propose a new technique, the 2-level branch predictor selection mechanism, which uses more run-time information to improve the performance of the predictor selection mechanism.
>
> 4.1 2-level Branch Predictor Selection Algorithm
>
> It is now well-known that the Two-Level Branch Predictor improves prediction accuracy over previously known single-level branch predictors [3]. The concepts embodied in the Two-Level Predictor can also be applied to the hybrid branch predictor selection mechanism. Figure 2 shows the structure of the 2-level predictor selection mechanism. A Branch History Register (BHR) holds the branch outcomes of the last m branches encountered, where m is the length of the BHR. This first level of branch history represents the state of branch execution when a branch is encountered. No extra hardware is required to maintain the first level of history if one of the component predictors already contains this information. That is, if the component predictor maintains a BHR, then the 2-level BPS mechanism does not need to maintain another copy of the BHR; instead, it just uses the component predictor's BHR. The Branch Predictor Selection Table (BPST) records which predictor was most frequently correct for the times this branch occurred with the associated branch history. This second level of history keeps track of the more accurate predictor for branches at different branch execution states.
>
> When a branch is fetched, its instruction address and the current branch history is used to hash into the BPST. The associated counter is then used to select the appropriate prediction. By using the branch history to distinguish more execution states, 2-level predictor selection scheme can more accurately select the appropriate predictions.
>
> Since the BPST and the PHT can be accessed in parallel, the time required for a hybrid predictor to make a prediction is max (BPST_time, PHT_time) + mux_time where BPST_time, PHT_time, and mux_time are the access time of BPST, PHT, and mux respectively. Since the PHT are often much larger than the BPST, PHT_time is greater than BPST_time. Thus, the time for making a prediction is PHT_time + mux_time, which is almost equivalent to the access time of single-scheme predictors.

Applicants note, similar to above, the first paragraph of the cited section is directed towards evaluating performance and efficiency. The second paragraph of the cited section discloses the manner, in which a 2-level predictor selection scheme selects predictions by using a Brach Predictor Selection Table (BPST) to determine which predictor is most frequently correct. The

third paragraph discloses, similar to above, the use of the counter to select the appropriate prediction mechanism. The last paragraph describes the calculation of the time required to make a prediction.

Applicants submit that throughout these cited sections and in the Patt reference as a whole, there is no teaching suggestion or disclosure of at least "a meta predictor to receive as inputs an index value and a branch prediction *to generate a misprediction value*" as described in embodiments of the present inventions. Indeed, Applicants submit there is no disclosure of any value that remotely resembles a misprediction value as described in embodiments of the present application. By way of background, further support for this limitation can be found at least at page 6 line 23 of specification which states:

> Misprediction value 112 then may be used *to decide whether to reverse the prediction provided by the base predictor*, or branch prediction 108.

The Patt reference does not contain any such reference, teaching or suggestion to the ability to reverse the prediction provided by a base predictor.

The Examiner argues that though described in the specification, Applicants' effort to further clarify "misprediction value" of the embodiment of claim 1 as having amongst other things "the ability to reverse the prediction provided by a base predictor" is improper. Applicants respectfully disagree and submit that these limitations were not meant to be read in to the claims. As argued above, the claim language, specifically the term "misprediction value", more than sufficiently describes an operation that a) is different from anything cited and b) is not disclosed in any of the cited references. Any such explanation taken from the specification is intended to describe and clarify one functionality of the "misprediction value", and to further clarify that any such functionality is not disclosed in the cited sections.

Although the McFarling reference is directed to towards combining branch predictors, it too is generally directed towards use of counters to enhance prediction performance, and does not disclose a misprediction value as described in embodiments of the present application anywhere. Therefore, Applicants submit that Mcfarling fails to make up for the deficiencies of Patt.

Therefore, Applicants argument remains, as affirmed by the cited sections, that the Patt or McFarling reference does not disclose a "misprediction value" as specifically recited in the embodiment of claim 1. In order to serve as the basis of a proper § 103(a) rejection, each and every element of a claim must be taught, suggested disclosed in the reference. Therefore, since each and every element of independent claim 1 is not taught, suggested or disclosed by the cited references, the §103(a) rejection is lacking and should be withdrawn. Independent claims 10, 17, and 20 contain substantively similar limitations and therefore should be allowed as well. Claims 2-8, 11-15, 18-19 and 21-23 depend from the aforementioned allowable independent claims, and therefore are in condition for allowance as well.

A. Claims 24-26 are not anticipated under 35 U.S.C. 102(b) as being taught by Tran (U.S. Patent No. 5,822,575)

The Office Action also alleges that Tran has taught restoring a base misprediction history register at column 14 lines 14 to column 15, line 7; column 18, lines 44-62; column 19, lines 31-49; Figure 3; and Figure 4). Applicants respectfully submit that there is no teaching suggestion or disclosure in the extensive citation of Tran of a "base misprediction history register" as disclosed in the embodiment of claim 24.

Column 14 lines 14 to column 15, line 7 generally describes a misprediction branch tag and a comparator block. The Office Action asserts that the branch tag shift is the equivalent of the "base misprediction history register", as the register's data is changed based upon the misprediction tag. Applicants disagree. First, as is well known in the art a tag is an instruction that specifies how a document is to be formatted (e.g., as in HTML). In light of this, Applicants submit that the formatting aspect inherent in the operation of this branch tag shift of Tran has no resemblance to a base misprediction history register.

For example , column 14, lines 16-18 state: "[a] misprediction is determined if the branch tags match and the prediction does not match the execution result". Clearly, this supports Applicants assertion that the branch tags of Tran are separate from an actual prediction. Furthermore, and more importantly, Applicants submit such a branch tag shift does not disclose a base misprediction history *register* as disclosed in embodiments of the present invention. For similar reasons, Applicants submit the branch *tag shift register* 50 disclosed in Tran deals in formatting history and synchronization, and does not deal in misprediction histories.

The next cited section, column 18, lines 44-62 discloses the operation of a "a global shift register" that stores the most recent N taken/not taken *branch* predictions. *See* column 18, lines 3-4. However, even if the Examiner alleges that the "global shift register" is the equivalent of a "*branch* misprediction history register" (which it is not), Applicants submit that it is not a "*base* misprediction history register" as recited in embodiments of the present application. They are two separate aspects of embodiments of the present application (e.g., independent claim 24). By way of background, support for the "base misprediction history register" can be found, among other places, at page 6 lines 16-20 of the specification, which state:

> As discussed above, base misprediction history register 110 reflects the correctness of the base predictor standing alone. Unlike global history registers that record whether previous branches were taken or not taken, base misprediction history register 110 records whether previous branch predictions were correctly predicted by the base predictor.

Therefore, in order to be a proper rejection, Applicants submit the Tran reference must disclose a *base* misprediction history register as found in embodiments of the present application.

Lastly, column 19, lines 31-49 disclose a control unit to reset valid bits for each branch tag assigned to an instruction is more recent (in program order) than the branch instruction having a branch tag of "8". Applicants submit that the "control unit" does not resemble a register at all, and therefore is inadequate to support a proper rejection.

The Office Action further argues that though described in the specification, Applicants' effort to further clarify the "base misprediction history register" of the embodiment of claim 1 by introducing sections of the specification was improper. Applicants respectfully disagree and submit that these limitations were not meant to be read in to the claims. As argued above, the claim language, specifically the term "base misprediction history register", more than sufficiently describes an operation that a) is different from anything cited and b) is not disclosed in any of the cited references. Any such explanation taken from the specification is intended to describe and clarify one functionality of the "misprediction value", and to further clarify that any such functionality is not disclosed in the cited sections. Applicants argument remains that the base misprediction history register or any equivalent thereof has not been disclosed in Tran.

Therefore, Applicants submit that none of the cited section of Tran disclose the "base misprediction history register" found in embodiments of the present invention. Therefore, Applicants submit that branch tag shift concept of Tang cannot properly serve as the basis of a

102 (b) rejection of claim 24. Claims 25-27 depend from the aforementioned allowable independent claims, and therefore are in condition for allowance as well.

Appellants therefore respectfully request that the Board of Patent Appeals and Interferences reverse the Examiner's decision rejecting claims 1-27 and direct the Examiner to pass the case to issue.

The Examiner is hereby authorized to charge the appeal brief fee of **$500.00** and any additional fees which may be necessary for consideration of this paper to Kenyon & Kenyon Deposit Account No. **11-0600**.

Respectfully submitted,

KENYON & KENYON

Date: December 30, 2005 By: /Sumit Bhattacharya/
Sumit Bhattacharya
(Reg. No. 51,469)

KENYON & KENYON
333 West San Carlos St., Suite 600
San Jose, CA 95110
Telephone: (408) 975-7500
Facsimile: (408) 975-7501

APPENDIX

(Brief of Appellant Stephan J. Jourdan et al.
U.S. Patent Application Serial No. 09/749,405)

8. CLAIMS ON APPEAL

1. (Previously presented) A branch prediction apparatus, comprising:

a base misprediction history register to provide an output;

a meta predictor to receive as inputs an index value and a branch prediction to generate a misprediction value in accordance with said inputs and said base misprediction history register output; and

a logic gate to receive said branch prediction and said misprediction value to generate a final prediction.

2. (Original) The branch prediction apparatus of claim 1, wherein said base misprediction history register includes misprediction history data.

3. (Original) The branch prediction apparatus of claim 1, further comprising an instruction that provides said index value.

4. (Original) The branch prediction apparatus of claim 3, wherein said instruction is a branch instruction.

5. (Original) The branch prediction apparatus of claim 4, wherein said final prediction determines a branch for said branch instruction.

6. (Previously presented) The branch prediction apparatus of claim 1, further comprising a branch predictor that receives said index value and generates said branch prediction.

7. (Original) The branch prediction apparatus of claim 6, wherein said branch predictor utilizes a prediction scheme to generate said branch prediction.

8. (Original) The branch prediction apparatus of claim 6, wherein said branch predictor includes a target address field and a prediction table.

9. (Original) The branch prediction apparatus of claim 1, wherein said base misprediction history register contains values of zero (0), and the misprediction value is not generated by said meta predictor.

10. (Original) A method for predicting branches, comprising:

receiving an index value, a branch prediction value correlating to said index value, and a misprediction history value at a meta predictor; and

generating a misprediction value at said meta predictor.

11. (Original) The method of claim 10, further comprising generating said branch prediction value at a branch predictor.

12. (Original) The method of claim 11, further comprising receiving an index value at said branch predictor.

13. (Original) The method of claim 10, further comprising generating a final prediction according to said branch prediction and said misprediction value.

14. (Original) The method of claim 10, further comprising determining a final value, and updating said meta predictor and said base misprediction history register according to said final value.

15. (Original) The method of claim 14, wherein said updating includes comparing said final value to said branch prediction.

16. (Original) The method of claim 10, further comprising bypassing said meta predictor when said misprediction history value contains all zeros (0).

17. (Previously presented) A processor, comprising:

a branch predictor to generate a branch prediction;

a base misprediction history register;

a meta predictor to receive an index value, said branch prediction and base misprediction history register data to generate a misprediction value.

18. (Original) The processor of claim 17, further comprising a final prediction to correlate to said misprediction value and said branch prediction value.

19. (Original) The processor of claim 17, further comprising a logic gate to generate said final prediction.

20. (Original) A computer readable medium having stored a plurality of executable instructions, the plurality of instructions comprising instructions to:

receive an index value, a branch prediction value correlating to said index value, and a misprediction history value at a meta predictor; and

generate a misprediction value at said meta predictor.

21. (Original) The computer readable medium of claim 20, further comprising an instruction to generate said branch prediction value at a branch predictor.

22. (Original) The computer readable medium of claim 21, further comprising an instruction to receive an index value at said branch predictor.

23. (Original) The computer readable medium of claim 19, further comprising an instruction to generate a final prediction according to said branch prediction and said misprediction value.

24. (Original) A method for restoring a branch prediction apparatus following a branch misprediction of a branch instruction, comprising:

restoring a base misprediction history register; and

restoring a branch predictor history register.

25. (Previously presented) The method of claim 24, further comprising updating a branch predictor.

26. (Original) The method of claim 24, further comprising updating a meta predictor.

27. (Original) The method of claim 24, further comprising flushing an instruction pipeline processing said branch instruction.

9. EVIDENCE APPENDIX

No further evidence has been submitted with this Appeal Brief.

10. RELATED PROCEEDINGS APPENDIX

Per Section 2 above, there are no related proceedings to the present Appeal.